HW 2 Due on February 12th at 4:30 pm. Name:_________________________________

1. Bytes per dollar (Memory) 1pt

If a 300 GB hard drive costs $100 dollars and a 4.7 GB DVD disk costs $2 dollars, which is a better deal in terms of bytes per dollar? Show your work.

2. Estimate (Memory) 1pt

How many hard drives could you fit in a shoebox? How many USB thumb drives could you fit in a shoebox? Use rough estimates for the size of a hard drive, USB thumb drive, and shoebox and show your work.

How many bytes are stored in the shoebox full of hard drives? How much does it weigh?

How many bytes are stored in the shoebox full of USB thumb drives? How much does it weigh?

I am not looking for exact answers to this question! Show your work and if the number you get is reasonable based on your assumptions, you'll get full credit! (But don't try to tell me that you can fit 10 million USB thumb drives into a shoebox ....)

3. 32-bit SSN (Encoding) 1pt

Suppose that the government decided to assign every person in the U.S. a 32-bit number (a binary number of length 32) instead of assigning each person a 9-digit decimal number. Would any two people be associated with the same number? Justify your answer.

Hint: Suppose the government assigned every person a 2-bit number. There are four possible 2-bit patterns: `00`, `01`, `10`, `11`, so if the US had a population of four, no two people would be assigned the same 2-bit number.

4. Encoding I (Encoding) 1pt

The 7-bit Extended ASCII character set relates 128 binary numbers to 128 commonly used characters in the English language (technically, not all are characters, but ignore this fact). For example, `1100001` corresponds to the character "a" and `1000001` corresponds to the character "A".

The Chinese character set is composed of unique characters that taken together comprise the written Chinese language. A college-educated Chinese adult is fluent with 6,000 to 7,000 unique Chinese characters.

How many **bits** are required to represent the entire set of written Chinese characters for a college-educated Chinese adult? How may **bytes**?

5. Encoding II (Encoding) 1 pt

A forensic computer scientist finds the following list of binary values on a hard drive (spaces added to improve readability): `01001100 01001111 01001100`. Assuming that the information is encoded as 7-bit ASCII, what does this series of numbers represent?

A. :-)
B. :-(
C. LOL
D. Woof
E. oNaMoNaPiA

6. Bit range (Encoding) 1 pt

How many bits are required to write the binary representation of the decimal number 512?

A. 8
B. 9
C. 10
D. 11
E. None of the above

7. Digitizing measurements (Encoding) 2 pts

Extra credit.

Suppose that we wanted to store a number on a computer, but were only allowed to use two bits to represent each measurement. The four possible bit patterns of length two are `00`, `01`, `10`, and `11`.

When reading the bits from memory, we can tell the computer that when it encounters a `00` that this means the expression `red`. Therefore, if I wanted to save the list `red green green red magenta`, I could just write the bits `00 01 01 00 11`. Now when I read back the bits, I know that this person meant `red green green red magenta`. In a similar way, I can define these combinations of bits to mean numbers other than their decimal equivalent, two examples are shown in the "bit-pattern-to-meaning-table" shown to the right.

bit-pattern-to-meaning table

Bit pattern	Meaning 1	Meaning 2
00	red	0
01	green	-1
10	blue	+1
11	magenta	2

Suppose that your instrument can take on values of 0.0, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, and 4.0 and you want to represent each number as a group of three bits that form a "bit pattern". The first part of your "bit-pattern-to-meaning-table" would look like the table to the right.

bit-pattern-to-meaning table

Bit pattern	Meaning
000	0.0
001	0.5
010	1.0
011	1.5
100	2.0
...	...

7.1. Part I

Suppose that you only have the choice of using three bits. How many unique combinations of 1s and 0s are possible?

7.2. Part II

Suppose that you get a new instrument that can measure values from 0.0, 0.1, 0.2, ..., 4.0, but you still store

data using three bits and use the same "bit-pattern-to-meaning-table".

When you write a bit group to record a measurement, you chose the bit grouping that has the closest value to your measurement. In doing so, you lose precision much like when you round a decimal number 1.11111111 to 1.1. When you return to the number 1.1, you won't be able to tell if the original number was, for example 1.11 or 1.12, because the both round to 1.1.

When you read back your measurements after representing each number on your disk as a group of three bits, what will be the largest difference between the read back value and the actual measured value? **Explain your answer in words.**

A. 0.1
B. 0.5
C. 0.4
D. 1.0
E. 2.0